Exploring Linear Discriminant Analysis Classification of Non-Normal Data using Poker Hands

Nicholas Zuber and Lucas Tata

University of Massachusetts Lowell
{nicholas_zuber, lucas_tata}@student.uml.edu

Abstract. Using linear discriminant analysis (LDA) as a linear classifier can produce well behaved results when the assumptions of this method are met. One of the core assumptions made in LDA is that the data being used is normally distributed. When this particular assumption is violated, it's known that an LDA classifier will produce inconsistent results that vary in correctness. In this paper, we explore this phenomenon and analyze the nature of the data's correspondence with its results. We first approach this problem by constructing an LDA model and investigating the distribution of features from a dataset consisting of enumerated poker hands. This dataset is briefly juxtaposed with another dataset that contains normally distributed data in order to emphasize how the two types of data differ due to the normality of its distribution. We then use this information to further explore potential approaches that could help optimize our results for accuracy, and the implications of said approaches.

1 Introduction

We explore the effectiveness of linear discriminant analysis on a multi-class non-normally distributed dataset. We expect the initial results of running this algorithm on such data will result in high error due to the violations that the LDA algorithm makes. We analyze these results and contrast normally distributed results in efforts to fully understand the characteristic differences between the data. We then perform artificial adjustments to both the non-normal data and our LDA classification method in an attempt to reduce the error. These adjustments include feature scaling and a comparison with a quadratic discriminant analysis model.

This is an interesting topic to analyze because as it stands currently within the Machine Learning community, contemporary implementations of LDA with normally distributed data will result in reduced error and high accuracy, but with non-normally distributed data the results are inconsistent. This analysis will aim to shed some light on this topic, uncover reasons why the results are so inconsistent, and prove if there are ways to artificially modify the data in such a way to make current LDA models more consistent.

We first analyze the independence and distributions of our feature set within our training data. From looking at the relative scatter between each class for any given feature alongside the respective distributions, we strengthen our assumptions that our dataset of poker hands is non-normal. Afterwards, we create LDA classifier with our non-normal training data. We take that classifier and test it with our test set, and output a final accuracy for our classification model. Then, we take that same LDA classifier and run it with normally distributed training and testing data. In comparing the output accuracy, we will then go more in depth analyzing the differences between the two data inputs. To further emphasize our reasonings for our results, we perform a dimensionality reduction on both our non-normal and normal datasets using our LDA model. Based on this analysis, we will attempt to change various characteristics about our data and our classification model. We believe this is an interesting approach to LDA, and yields insightful results.

Our non-normally distributed dataset is the UCI poker hand dataset, and it can be accessed freely from all parties at UCI's official website [Cat07]. Similarly, our normally distributed data set is the UCI Wine dataset, and it can also be accessed freely from all parties at UCI's official website [For91].

2 Background

We have searched through the current research of the Machine Learning community, and we could not find a source that performed an analysis of LDA on non-normally distributed data. Our research presented here will hopefully serve as the start of a discussion on challenging preconceived notions about LDA.

There has been previous work on a similar poker hand data set to the one we use in our research, but the actual research does not pertain to our topic. There has been some research on LDA in general. Some notable LDA research we came across included research by Jieping Ye of Arizona State University who researched least square linear discriminant analysis[Ye07]. This research was interesting to us because Jieping explores applying an algorithm that is not usually associated with multi-class LDA, and this challenges the machine learning community's perception on LDA.

We believe our research also challenges the perceptions about LDA, and we aim to shed light on some assumptions made when running linear discriminant analysis. Previously, research done on linear discriminant analysis is more geared towards the comparison between LDA and other classification methods. Whenever LDA is performed, assumptions are made that the input data set is normally distributed. We believe if we challenge these norms and analyze why these assumptions are made and explore other options to circumvent these assumptions we will pave a way for the community to challenge other machine learning norms.

3 Approach

The first step to our approach is to create a linear discriminant analysis model from our non-normal poker hand data set. Our poker hand data set has both a training and validation set. We first read in our training and validation sets, and parse our data into our feature data, and the corresponding class data. We calculate the prior probabilities of each class occurring in the data set, their respective means, and compute the pooled covariance matrix of the input data. We then calculate the discriminants for each class with the training data given by the decision function [Bis06]:

$$\delta_k = \left[x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(\pi_k) \right] \tag{1}$$

Once all of the determinants are calculated, the predictions of the LDA model are given by the maximization of the decision functions for each class [Bis06]:

$$\hat{G}(x) = \arg\max_{\text{k}} \delta_k(x) \tag{2}$$

We then take all of our calculated predictions and analyze the accuracy and correction for the entire model and each respective class.

In order to verify our approach to LDA is correct, we follow the same process to create and test a model with the normally distributed wine data set. Our resultant accuracy for both the poker hand and wine data set is theoretically going to be vastly different, with the assumption that the normally distributed data will have low test error and will serve as our control.

We analyze the difference between the data sets by performing a dimensionality reduction on each. The purpose of dimensionality reduction is to allow us to visualize the data by optimizing the separability between all of the different classes, while preserving the class discriminatory information. Based on this analysis, we explore ways to reduce the error in the data. We explore feature scaling the data before we perform LDA, by applying offset values to the prior probability of the classes:

$$p(C_i) = p(C_i) \pm \Delta_k \tag{3}$$

We also explore changing how the covariance matrices are calculated with our LDA model. LDA typically uses a common pooled covariance matrix between the classes. Instead, we used a single covariance matrix for each class. We also calculated a discriminant for each feature per class, instead of calculating a discriminant for an entire class. We believe this can yield better results due to the fact that even though the data is poorly distributed, could have higher accuracy because it does not pool the features together in such a way that our original LDA method does.

We run this modified version of LDA on our non-normally distributed data by computing the prior probabilities, the class means, and covariance matrices. We then calculate the discriminants for each feature per class with this equation:

$$\delta_k^{'} = \left[-\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + log(\pi_k) \right] \tag{4}$$

We then follow the same classification guesses as:

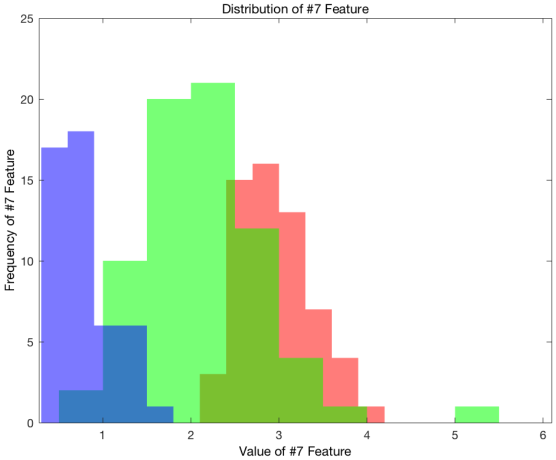
$$\hat{G}'(x) = \arg\max_{\text{k}} \delta_k'(x) \tag{5}$$

4 Results

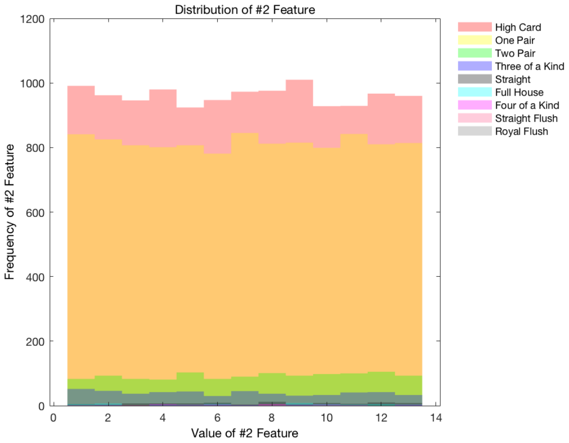
Throughout our entire analysis, we use exactly two different public datasets provided by the UCI machine learning repository. As mentioned earlier, these particular datasets include a enumerated set of poker hands and a chemical analysis of wines. As a reminder, both of these datasets can be accessed publicly for free at the UCI official website.

Our primary focus is on the dataset of poker hands, since this is our non-normally distributed experimental group. The dataset consisting of wines and their respective chemicals is used a Gaussian control group which provides relevant context to what is generally expected for an LDA model in terms of data assumptions.

Before we began with our linear discriminant analysis, we first want to make it clear that our poker hand dataset is non-normally distributed and what we mean exactly when we say this. When we refer to our dataset being Gaussian or non-normal, we are referring to the distributions of independent features. As a brief reminder, some data that is considered to be Gaussian or normally distributed will generally display some sort of bell-shaped curve when it is graphed. For the sake of simplicity and clarity, we will display the distribution of a single feature from our datasets which conveys the general distribution of all other features. First, let us consider the distribution of a single feature among the three classes from our wine dataset:



There is an overt bell-curve for each of the three classes for this feature. Also, we notice that there is a relatively good amount of separability as well, which is a good indicator that a linear discriminant model will be able to maximize this separability to better distinguish classes from one another. Now, we take a look at the distribution of a single feature from the poker hand dataset:



Here, you can clearly see the lack of any kind of bell-curve among the distributions. This also comes with another implication – there is no separability among the classes since all of the data is essentially equivalently spread among the classes. This helps us understand and visualize the non-normality of our dataset in juxtaposition to a Gaussian dataset.

Next, we create our LDA classification model. For this model, we essentially create a decision function that picks a class which has the maximum posterior probability given some input feature vector. We then proceed to maximize this function for all classes in our dataset in order to find the class which is most likely to belong with our input feature vector. Recall that this function look like

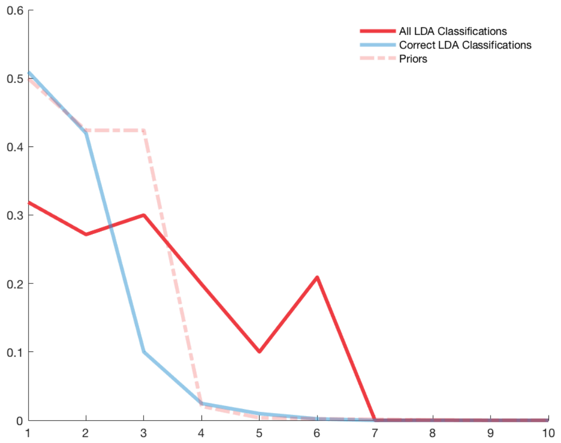
$$\hat{G}(x) = \arg\max_{\text{k}} \left[x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(\pi_k) \right] \tag{6}$$

We trained our model with approximately 25,000 poker hands and then ran tests on a separate set of 1,000,000 poker hands. The results were as expected – there was a low accuracy percentage of 28.164%. The more interesting part to us and our experiment is understanding why this accuracy rating was so low.

We then looked at the distribution of how these poker hands were classified. For reference, the input vectors were classified 318,729 times as *high card*, 271,553 times as *two of a kind*, 37 times as *full house*, 2 times as *royal flush*, and so on. This kind of distribution among the respective poker hand classes makes sense to us intuitively because of the calculated priors from these classes. Since our data here is not normally distributed, we can expect that our decision function δ_k will not be affected relatively as much by the input feature vectors. We had predicted this from our aforementioned feature distribution graph showing us an extremely low pattern of covariance between each class' features. This would inherently give the prior more weight when it comes to deciding the class as per our decision function definition.

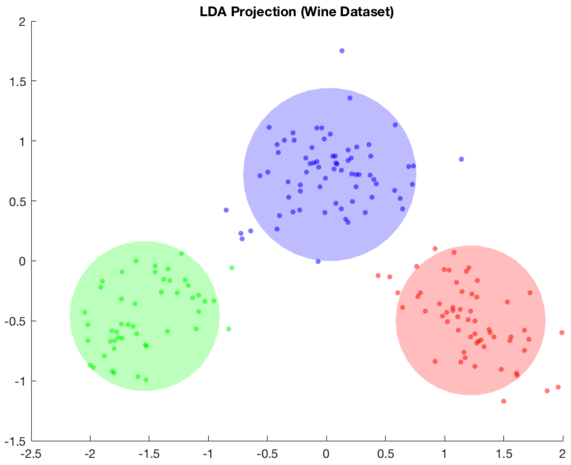
The next aspect we want to explore is how accurate each of these classifications are. As per our assumption and the results we've seen so far, we expect these results to be approximately as accurate as the class' respective prior. This is because we recall our aforementioned distributions of the poker hand features, and remember that there is almost no separability among the classes. This implies that there is no real accurate decision boundary which can be drawn from our data. So when a class is being determined by our LDA model from some input feature vector, the probability of it being classified correctly is relative to the probability of that class being selected in general.

The amount of influence the priors have on our LDA classification compared to the covariance counterparts can be visualized with this graph, displaying the percentage of all hand classifications, percentage of correctly identified hand classifications, and the respective priors:

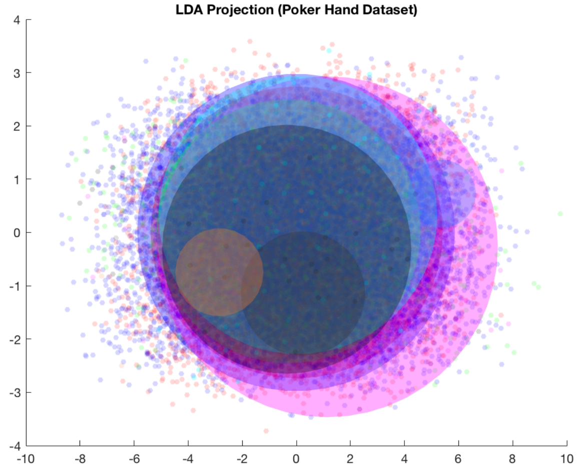


We then want to better understand what our LDA model is doing exactly when it tries to create these classifications. We know from our discriminant equations what's happening when probability is being calculated, but it would be helpful to also see visually how classes are optimized when the feature dimensions are reduced.

This is done by using our LDA model to maximize the difference in means by the difference in scatter between all of the classes in our dataset. Once this is done and maximized, we essentially create a new axis for the data points to be projected onto such that the difference between classes is optimized for separability. When this is done correctly with normally distributed data, we produce a graph with clear lines of separability between each class. For example, when we perform this LDA dimensionality reduction algorithm on our normally distributed wine dataset, we get something like this:



On this graph, you can clearly see the separation of classes such that there could exist some decision boundaries to reasonably classify input vectors correctly. Now, when we perform the same algorithm with our non-normally distributed poker hand dataset, we end up with a graph that looks like this:



Here, there is a large amount of overlap between almost all of the classes. From our testing and analysis, we understand this to make sense because there was almost no separability between the poker hand features to begin with, therefore when we attempt to maximize the mean and scatter ratios, there isn't much we can maximize.

So knowing all of this information on why LDA models give relatively poor results with non-normal data, we then attempt to impose some techniques to improve on the results.

The first thing that we do is add a feature scale to our classes. The particular scale we used here was adding a constant to our probabilities which was consistent with the actual probability of the respective class occurring in a game of poker. We saw after doing this that there was very little improvement on our results, which made sense because we understand that our source of error was coming from our very low covariance. This cannot be countered with a feature scale in this regard.

Knowing that we need to improve on the covariance within our model, we next tried to modify the way our model calculates the covariance. Instead of using the traditional pooled covariance among all of the classes, we calculated individual covariance matrices for each of the respective classes. We also calculated discriminants for each individual feature of each individual class, rather than simply having only a single discriminant for each class itself. After running the model with these adjustments, we noticed that we did not receive better results. This makes sense since with this particular approach, each feature is essentially considered in isolation per class, and with our dataset of poker hands, this would bare even less meaning for the feature's respective class.

5 Conclusion

Linear discriminant analysis classification produces accurate results when data is normally distributed. We purposely violated this assumption by using a non-normally distributed data set. When we ran our LDA classifier on the non-normally distributed data, our resultant model had high test error. Our attempts to compensate for our poorly structured input data set included feature scaling and using quadratic discriminant analysis in hopes it would reduce the error.

Our attempts did not reduce error significantly, but we displayed that through analysis of the input data, you are able to slightly overcome assumptions made for linear discriminant analysis. From our findings and deeper understanding of why LDA fails on non-normal data, It is possible in the future that there will be a more definite way to compensate for the poor distribution of input data, which will allow the community to run linear discriminant analysis on any data set resulting in low test error.

References

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer Science and Business Media LLC, 223 Spring Street, New York, New York 10013, 2006.

[Cat07] Robert Cattral. UCI poker hand data set, 2007.

[For91] PARVUS Forina, M. et al. UCI wine data set, 1991.

[Ye07] Jieping Ye. *Least squares linear discriminant analysis.* Arizona State University, Tempe, AZ, Corvalis, Oregon, USA, 2007.

Lucas Tata & Nicholas Zuber: Both members of the team contributed to the different facets of this research equally and as a group.